

UNIVERSIDAD CATÓLICA BOLIVIANA "SAN PABLO"
UNIDAD ACADÉMICA REGIONAL COCHABAMBA
Departamento de Ingeniería y Ciencias Exactas
Internet de las Cosas



Integración de un dispositivo inteligente con una plataforma en la nube (AWS) con la ayuda del protocolo de aplicación MQTT

Integrantes: Denilson Fiesta Inocente
Jonathan Moruno Teran
Docente: Eduardo Enrique Marin Garcia

Cochabamba - Bolivia
3 de Junio de 2024

I. INTRODUCCIÓN

En esta era de la digitalización, la tecnología se ha hecho cada vez mas presente en nuestra vida cotidiana, desde los dispositivos inteligentes en nuestros hogares hasta las ciudades inteligentes. El internet de las cosas (IoT) ha sido crucial para esta transformación, revolucionando nuestra interacción con el entorno que nos rodea.

El IoT se refiere a la transformación de los objetos físicos para dotarlos de inteligencia y la capacidad de conectarse a internet. Un objeto inteligente es aquel que cuenta con tecnología integrada, capaz de percibir e interactuar con su entorno. Estos objetos pueden comunicarse entre sí a través de internet, lo que les permite recopilar y compartir datos para automatizar tareas, mejorando la eficiencia y la calidad de vida.

Con la creciente presencia de estas tecnologías en nuestra sociedad, es esencial comprenderlas adecuadamente. Por esta razón, el objetivo del presente informe es diseñar e implementar un circuito funcional para un dispensador de agua automático, utilizando una celda de carga de 1 kg, un modulo HX711, un puente H, una mini bomba de agua sumergible, una placa ESP32 y el asistente de voz de Alexa. Se detallara el proceso de desarrollo del sistema, desde el diseño inicial hasta la implementación y las pruebas realizadas para verificar su funcionamiento.

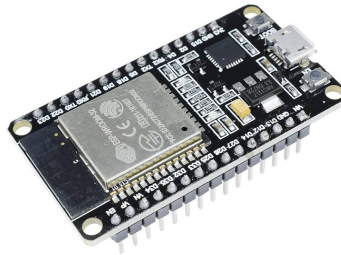
II. MARCO TEÓRICO

A continuación, se detallan los objetos inteligentes utilizados en la construcción del circuito y las herramientas de software que manejan la lógica funcional del proyecto:

■ MÓDULO ESP32

El modulo ESP32 es un microcontrolador con un procesador de doble núcleo y conectividad Wi-Fi y Bluetooth integrada. Actúa como el controlador central en varios sistemas, coordinando la comunicación entre actuadores y sensores. Se manipula mediante un cable micro USB conectado a una computadora y un IDE apropiado (TEJA, 2024).

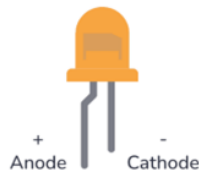
Figura 1
MÓDULO ESP-WROOM-32



■ DIODO EMISOR DE LUZ

Un LED es un componente electrónico que emite luz cuando se le aplica corriente eléctrica en la dirección adecuada. Usualmente, se compone de 2 pines: el ánodo y el cátodo. El ánodo se conecta al voltaje positivo, mientras que el cátodo se conecta al negativo. Es bastante susceptible al daño por corriente excesiva, por lo que se recomienda combinarlo con un resistor para controlar la corriente que circula a través de él (HARRIS, 2024)

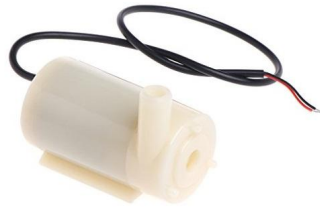
Figura 2
LED Y SUS COMPONENTES



■ MINI BOMBA DE AGUA SUMERGIBLE

Una mini bomba de agua sumergible es un dispositivo compacto diseñado para el bombeo eficiente de agua en diversas aplicaciones. Dicho dispositivo permite impulsar el agua que recibe en la entrada, obteniendo una salida de aún mayor presión. Cuenta además con un caudal de bombeo entre 80 y 120 litros por hora (UNIT ELECTRONICS, 2018).

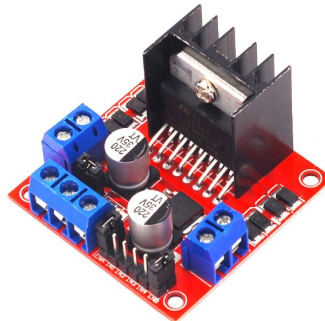
Figura 3
MINI BOMBA DE AGUA SUMERGIBLE



■ DRIVER PUENTE H L298N

El driver puente H L298N es el módulo más usado para el control de motores de corriente directa. Este nos permite mediante señales digitales el control de la velocidad y el sentido de giro de motores, que claramente están integrados en bombas de agua. Además, su diseño robusto y su capacidad para manejar corrientes de hasta 2 amperios lo hacen ideal para una amplia gama de aplicaciones en robótica y automatización (NAYLAMP MECHATRONICS, 2023).

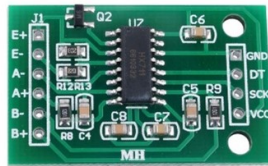
Figura 4
DRIVER PUENTE H L298N



■ MODULO HX711

EL módulo HX711 es un convertidor analógico digital de 24 bits diseñado específicamente para básculas electrónicas. El módulo HX711 se utiliza para convertir señales analógicas provenientes de sensores en valores digitales. Se caracteriza por su alta precisión, bajo consumo de energía y facilidad de uso (ALLDATASHEET, 2024).

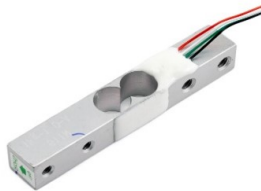
Figura 5
MODULO HX711



■ CELDA DE CARGA DE 1 KG

La celda de carga de 1 kg es un componente fundamental en básculas y sistemas de medición de peso. Se trata de un dispositivo que convierte la fuerza aplicada sobre ella en una señal eléctrica medible. El Transmisor HX711 está especialmente diseñado para trabajar con celdas de carga de este tipo y ofrecer una salida digital para ser leída por el microcontrolador. (NAYLAMP MECHATRONICS, 2023).

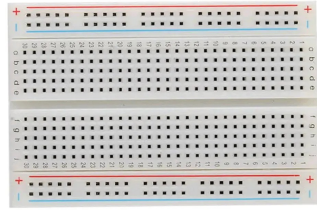
Figura 6
CELDA DE CARGA DE 1 KG



■ PROTOBOARD

Una protoboard es una placa de circuito impreso que se utiliza para realizar conexiones entre componentes electrónicos. Esta se divide en filas y columnas de agujeros interconectados eléctricamente, lo que facilita la conexión de componentes sin la necesidad de soldadura (PORTAL ACADÉMICO DEL CCH, 2021).

Figura 7
PROTOBOARD DE 400 PINES



■ **Alexa**

Alexa es un servicio de voz ubicado en la nube de Amazon, cuenta de igual forma con presencia en dispositivos terceros. Alexa permite la interacción con la misma a través de la voz ofreciendo una experiencia más natural a los clientes que la usan para interactuar con otros dispositivos. Alexa es capaz de integrarse a una gran cantidad de objetos inteligentes, pudiendo añadirles un sinnúmero de funcionalidades extras (AMAZON WEB SERVICES, 2024).

Figura 8
LOGO DE ALEXA



■ **AWS IoT CORE**

AWS IoT Core es un servicio en la nube proporcionado por Amazon Web Services (AWS) que permite la conexión de dispositivos IoT a los servicios de AWS. AWS IoT Core también dirige billones de mensajes a los servicios de AWS. Podemos realizar todo esto sin necesidad de administrar la infraestructura de este servicio (AMAZON WEB SERVICES, 2023).

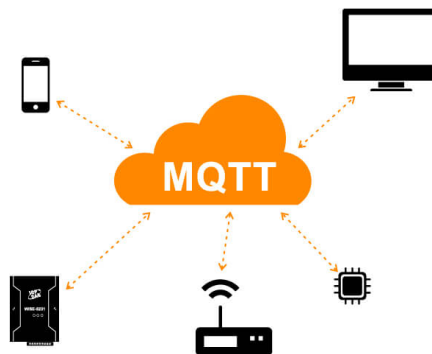
Figura 9
Logo AWS IoT CORE



■ MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería liviano. AWS IoT Core cuenta con alta compatibilidad con este protocolo, pudiendo admitir conexiones de dispositivos que utilizan el mismo no sin antes identificarse mediante un ID de cliente. MQTT sigue un modelo de comunicación de publicación-suscripción, donde los dispositivos publican mensajes en tópicos o temas y los suscriptores interesados reciben esos mensajes (MQTT, s.f.).

Figura 10
PROTOCOLO MQTT



■ AWS IoT CORE SHADOWS

Las sombras son un reflejo virtual del estado de un dispositivo. Algunas operaciones que los clientes web, otros dispositivos y servicios pueden realizar mediante el protocolo MQTT con sombras son: Su creación, actualización y eliminación mediante la consola de AWS IoT. Los objetos de AWS IoT no tienen sombras cuando se crean

sin embargo pueden llegar a tener varias (AMAZON WEB SERVICES, 2024).

■ **AWS IoT CORE RULES**

Las reglas otorgan a los dispositivos de AWS IoT Core con los que están relacionadas la capacidad de interactuar con los servicios de AWS. Estas usan mensajes MQTT que pasan por los protocolos de publicación o suscripción que sean compatibles con los protocolos de comunicación del dispositivo. Al comprender y aprovechar adecuadamente las reglas, se usa el potencial completo de AWS IoT Core para crear soluciones IoT innovadoras que resuelven problemas reales (AMAZON WEB SERVICES, 2024).

■ **AWS IoT ANALYTICS**

AWS IoT Analytics es un servicio de análisis de datos gestionado. Permite recolectar, procesar y almacenar datos de dispositivos IoT a gran escala. A partir de los datos generados por el mismo se pueden crear un sinnúmero de reportes a beneficio de los clientes que usan este servicio (AMAZON WEB SERVICES, 2023).

■ **AWS IoT CHANNEL**

Un channel en AWS IoT Analytics es un servicio que recopila y guarda datos de mensajes sin procesar los mismos. Ya con los mensajes guardados y sin procesar estos están listos para pasar por un pipeline. Los channels se pueden crear utilizando la consola de AWS IoT Analytics (AMAZON WEB SERVICES, 2024).

Figura 11
REPRESENTACIÓN VISUAL DE UN CHANNEL



■ **AWS IoT PIPELINE**

Un pipeline es simplemente una serie de actividades. Su función es procesar y filtrar mensajes antes de almacenarlos en un data store. Para que la información de un

channel llegue a un data store, el channel y el data store deben ser conectados a través de un pipeline (AMAZON WEB SERVICES, 2024).

Figura 12
REPRESENTACIÓN VISUAL DE UN PIPELINE



■ **AWS IoT DATA STORE**

Un data store es un repositorio de mensajes. Los data stores reciben y almacenan mensajes. Están diseñados para ser escalables, por lo que se puede almacenar en ellos tantos mensajes como se necesite (AMAZON WEB SERVICES, 2024).

Figura 13
REPRESENTACIÓN VISUAL DE UN DATA STORE



■ **CONSULTA SQL**

Una consulta SQL (Structured Query Language) es simplemente una instrucción en este caso a un data store. Permite a los usuarios realizar diversas operaciones sobre los datos almacenados. Las consultas SQL se componen de diferentes sentencias que especifican la acción a realizar, los datos a los que se aplica y las condiciones que deben cumplirse (W3SCHOOLS, 2024).

■ **AWS IoT DATASET**

Un dataset es un conjunto de datos. Su función es almacenar los resultados de consultas SQL ejecutadas en data stores. Las consultas SQL se pueden utilizar para filtrar, agregar y transformar los datos almacenados en un data store. (AMAZON WEB SERVICES, 2024).

Figura 14
REPRESENTACIÓN VISUAL DE UN DATASET

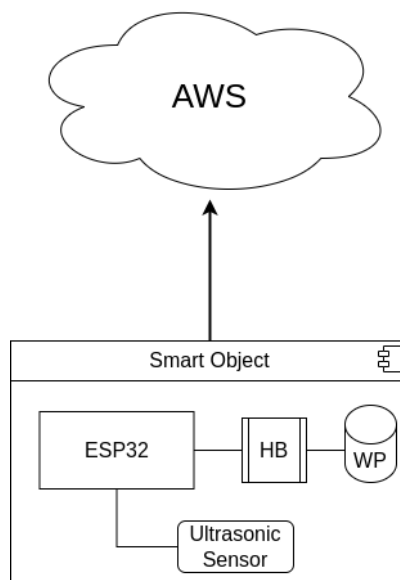


III. ANÁLISIS Y DISEÑO

Se ha diseñado un sistema de integración que asegura su correcto funcionamiento y cumple con los objetivos del informe. El sistema propuesto está compuesto por un módulos ESP32, una celda de carga, un modulo HX711, una mini bomba de agua, un puente H L298N, un protoboard de 400 pines y cables jumper.

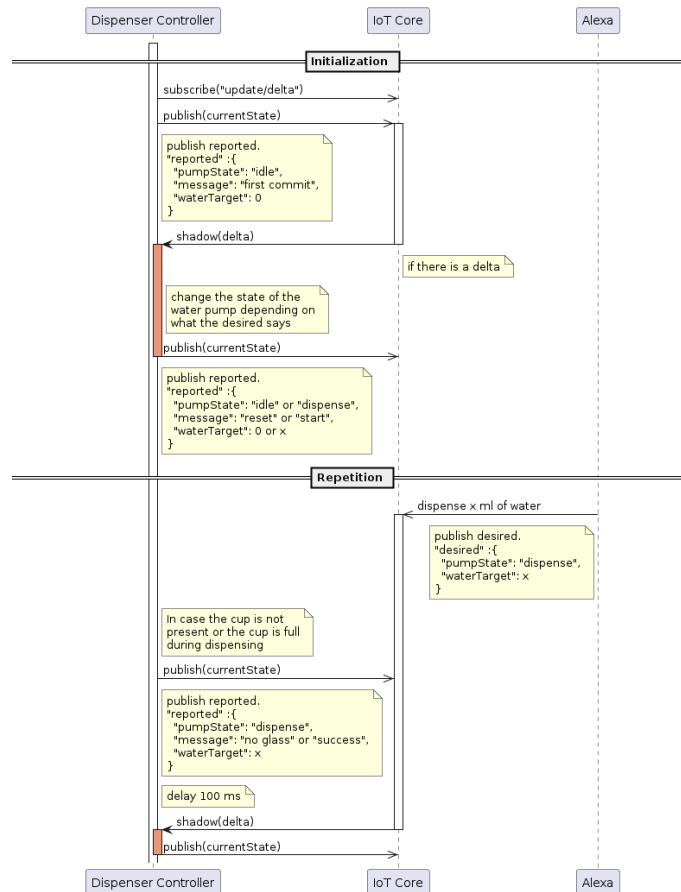
Inicialmente, se diseñó la estructura estática del sistema, que incluye los siguientes componentes: IoT Core, encargado de gestionar el manejo de datos del dispositivo inteligente; el Dispositivo inteligente, compuesto de un dispositivo actuador y un dispositivo sensor; Alexa, la asistente de voz encargada de gestionar las ordenes. Esta disposición puede ser visualizada en la **Figura 15**.

Figura 15
DIAGRAMA DE ESTRUCTURA DEL SISTEMA



Posteriormente, se llevó a cabo un análisis y diseño de la secuencia de interacciones entre los componentes esenciales del sistema, tales como el dispositivo inteligente, IoT Core y Alexa. Este modelo se encuentra representado de manera visual en la **Figura 16**.

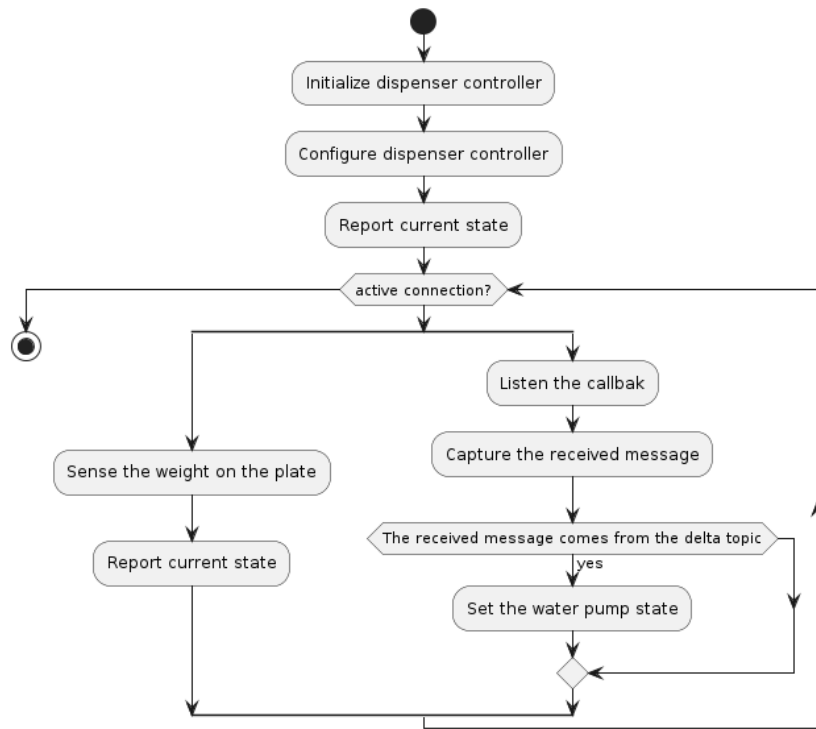
Figura 16
DIAGRAMA DE SECUENCIA DEL SISTEMA



Para una mejor comprensión de los procesos llevados a cabo por el dispositivo inteligente y IoT Core, se han elaborado los siguientes diagramas de flujo, presentados en las **Figuras 17, 18 y 19**.

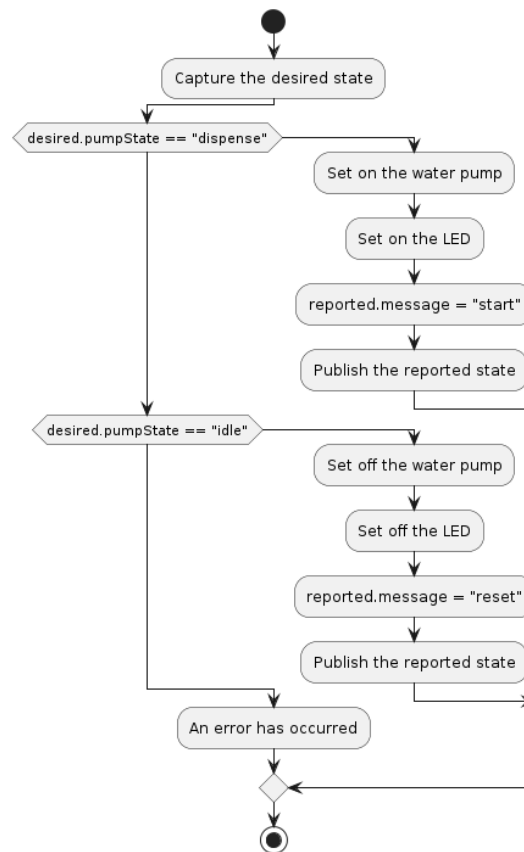
En la **Figura 17**, se detallan los procedimientos involucrados en el funcionamiento general del dispositivo inteligente.

Figura 17
DIAGRAMA DE FLUJO DEL DISPOSITIVO INTELIGENTE



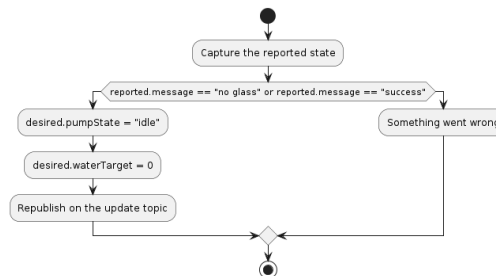
La **Figura 18** describe el proceso que sigue el objeto inteligente una vez que este captura un mensaje en el t3pico "update/delta", significando que existe un cambio de estados.

Figura 18
DIAGRAMA DE FLUJO DEL MANEJO DEL ESTADO DEL DISPENSADOR



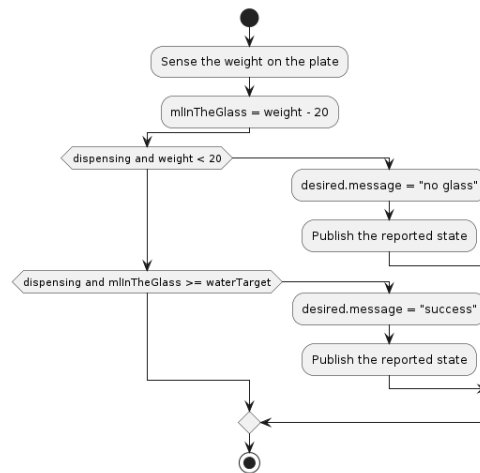
En la **Figura 19**, se muestra el proceso específico que se lleva a cabo en cada iteración del dispositivo inteligente. En esta función se realizan las lecturas del sensor.

Figura 19
DIAGRAMA DE FLUJO DEL MANEJO DE EXCEPCIONES EN EL LOOP



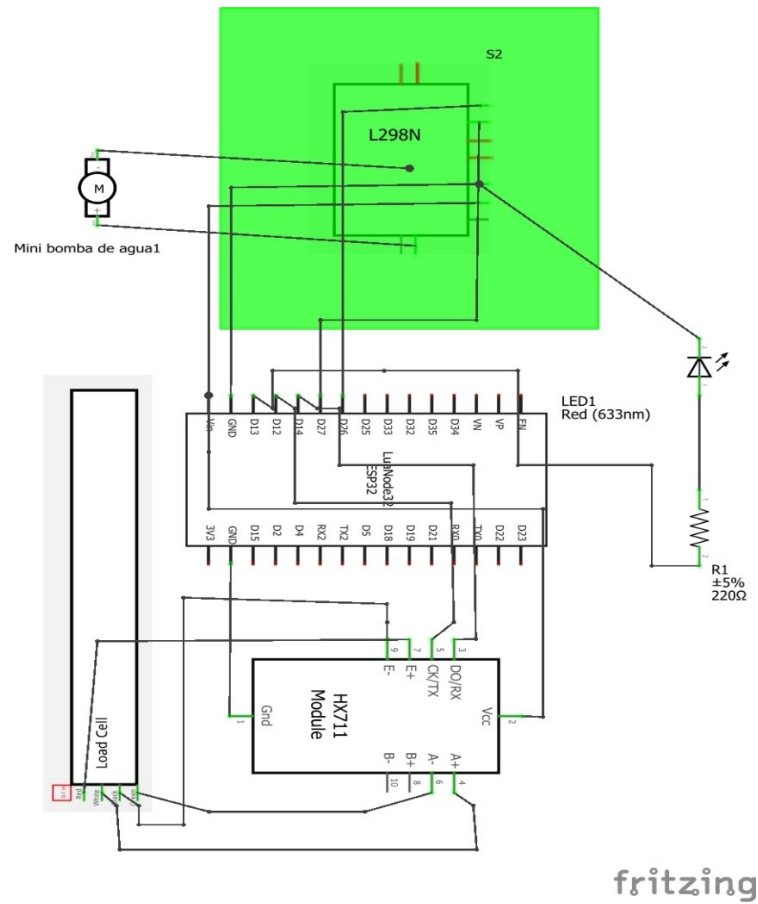
En la **Figura 20**, se muestra el proceso específico que la plataforma sigue cuando existen publicaciones en el tópico "update", esto realizándolo a través de las reglas de AWS IoT Core. Aclarar que se unieron la reglas bajo un mismo diagrama de flujo.

Figura 20
DIAGRAMA DE FLUJO DEL MANEJO DE REGLAS EN AWS



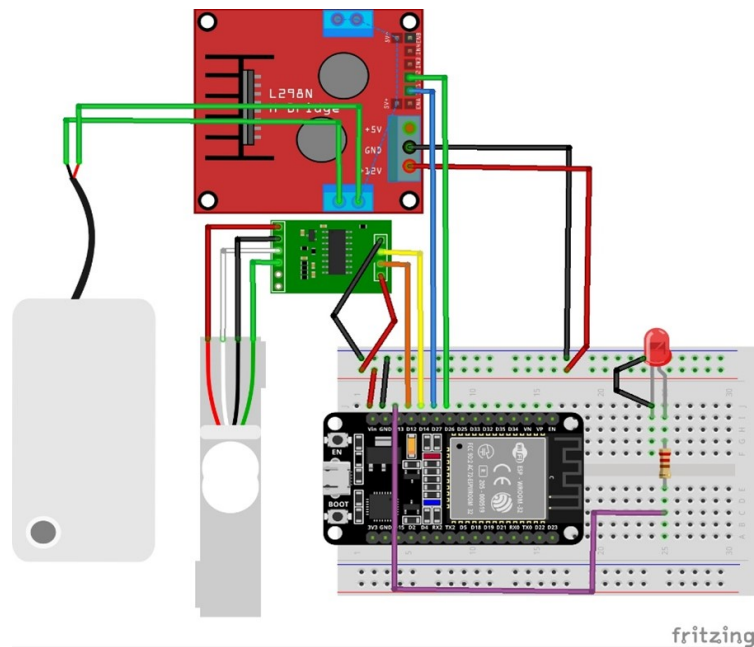
Posteriormente, se elaboraron diagramas de circuito para el dispositivo inteligente. El diagrama representa la disposición física y la conexión de los componentes electrónicos en el sistema. Este diagrama puede ser visualizado en la **Figura 21**.

Figura 21
DIAGRAMA DE CIRCUITO DEL DISPOSITIVO INTELIGENTE



Finalmente se muestra un prototipo de los componentes conectados. Este prototipo puede ser visualizado en la **Figura 22**.

Figura 22
PROTOTIPO DEL DISPOSITIVO INTELIGENTE



IV. DESARROLLO E IMPLEMENTACIÓN

Para lograr una funcionalidad completa del sistema, se programó el servidor TCP, la placa ESP32 del dispositivos actuador y sensor, con el objetivo de controlar la comunicación entre estos componentes a través de protocolos establecidos.

El código fuente utilizado en el presente informe se encuentra disponible en el siguiente enlace: <https://github.com/Al1bi/Practica4-IoT>.

El código fuente se desarrolló utilizando el paradigma de programación orientada a objetos (POO). Además, se siguieron convenciones de código, como la calistenia de objetos, con el objetivo de hacer el código fuente más legible y escalable.

A continuación se da una explicación breve de la siguientes clases:

- **AWSDispenserController.ino:** Controla el dispensador de agua, coordinando las acciones de dispensación según las instrucciones recibidas del AWS.

- **Dispenser.cpp:** Implementa la funcionalidad principal del dispensador, incluyendo la activación y desactivación del dispositivo. Trabaja en conjunto con el WeightSensor, WaterPump y Led.
- **WeightSensor.cpp:** Controla el sensor de peso utilizado para medir el peso de un objeto, proporcionando información para el control del dispensador.
- **WaterPump.cpp:** Controla la bomba de agua del dispensador, permitiendo su activación y desactivación según sea necesario.
- **Led.cpp:** Es un LED que se enciende cuando la bomba esta encendida.
- **MQTTManager.cpp:** Gestiona la comunicación mediante el protocolo MQTT, enviando y recibiendo mensajes entre el dispositivo y AWS.
- **MQTTConfig.cpp:** Responsable de configurar la conexión y los parámetros de comunicación para el protocolo MQTT utilizado en la interacción con el servidor.
- **AmazonCredentials.cpp** Se encarga de gestionar las credenciales necesarias para la autenticación y autorización en los servicios de Amazon Web Services (AWS).
- **WiFiManager.cpp:** Encargado de manejar la conexión WiFi del dispositivo, asegurando una conexión estable y confiable con el servidor TCP.
- **index.js:** Encargado de manejar la logica detras de los Intents de Alexa.

V. PRUEBAS Y VALIDACIONES

■ TIEMPO Y CAUDAL DE LLENADO DE UN VASO DE 130 ML CON MINI BOMBA DE AGUA

Para obtener el caudal y el tiempo en el que se llena un vaso de 130 mililitros con una mini bomba de agua se requirió de un cronometro. Esta herramienta nos brinda un tiempo en segundos con una precisión de dos decimales. Se registro un total de 30 datos obteniéndose la siguiente tabla, con un registro de tiempo en segundos por iteración, un promedio de tiempo de llenado y el caudal de la mini bomba de agua en litros por hora. Para más detalles respecto a cómo se obtuvieron los siguientes datos, véase este mismo punto en los **ANEXOS**.

Nro	t [s]
1	5.75
2	5.7
3	5.9
4	5.69
5	5.89
6	5.76
7	5.85
8	5.89
9	5.69
10	5.94
11	5.72
12	5.75
13	5.86
14	5.68
15	5.77
16	5.84
17	5.71
18	5.86
19	5.67
20	5.82
21	5.76
22	5.9
23	5.68
24	5.92
25	5.74
26	5.76
27	5.83
28	5.72
29	5.94
30	5.7
Tiempo de llenado promedio	5.78966667
Caudal [L/H]	80.83366918

Tabla1. Resultados obtenidos de tiempos de llenado de la mini bomba de agua

Fuente: Elaboración propia

■ FACTOR DE CALIBRACIÓN ÓPTIMO

Para determinar el factor de calibración más adecuado de la celda de carga utilizada en este proyecto, se empleó exclusivamente el monitor serial proporcionado por el entorno ARDUINO IDE y un objeto cuyo peso se conoce con precisión. El monitor serial registró el peso de un objeto de 209 gramos en doce mediciones sucesivas. Antes de iniciar el proceso, se configuró el factor de calibración en uno y se realizó un procedimiento de tara, el cual consiste en ajustar la celda de carga a 0 gramos

sin ningún objeto encima. Este paso es necesario para compensar el peso de los componentes adicionales que se encuentran sobre la celda de carga y que sirven de soporte para las mediciones. A continuación, se presenta la tabla de datos obtenidos durante las doce mediciones, las cuales permiten calcular el promedio del factor de calibración, identificado como el más óptimo para nuestro proyecto. Para detalles adicionales sobre la metodología utilizada para obtener estos datos, consulte el apartado correspondiente en los **ANEXOS**.

Numero	Lectura	Peso real	Factor de calibracion
1	52111	209	249,3349282
2	52225		249,8803828
3	52074		249,1578947
4	51885		248,2535885
5	52086		249,215311
6	52019		248,8947368
7	52158		249,5598086
8	51445		246,1483254
9	51981		248,7129187
10	51924		248,4401914
11	51729		247,507177
12	52463		251,0191388
Promedio			248,8437002

Tabla2. Resultados obtenidos del factor de calibración de la celda de carga de 1 kg

Fuente: Elaboración propia

■ ERROR RELATIVO PORCENTUAL DE CELDA DE CARGA DE 1 KG CON CALIBRACIÓN

Para determinar el error de una celda de carga calibrada de 1 kg, se utilizó el monitor serial del entorno ARDUINO IDE y un objeto de peso conocido, en este caso, un vaso lleno de agua con un peso preciso de 209 gramos. El monitor serial registró el peso de este objeto en 25 mediciones.

Antes de iniciar el proceso, se configuró el factor de calibración óptimo y se realizó un procedimiento de tara, el cual consiste en ajustar la celda de carga a 0 gramos sin ningún objeto encima. Este paso fue necesario para compensar el peso de los

componentes adicionales sobre la celda de carga, que sirven de soporte para las mediciones.

A continuación, se presenta la tabla de datos obtenidos de 25 mediciones, las cuales permiten calcular el promedio de la lectura calibrada del peso del vaso (209 gramos) y su respectivo error relativo porcentual. Para detalles adicionales sobre la metodología utilizada para obtener estos datos y el cálculo del error relativo porcentual, consulte el apartado correspondiente en los **ANEXOS**.

Numero	Lectura con calibracion
1	204,61841
2	204,57823
3	204,45767
4	204,36122
5	204,48178
6	204,51794
7	204,50188
8	204,46973
9	204,6144
10	204,71083
11	204,69878
12	204,73897
13	204,77916
14	204,92383
15	204,95999
16	205,11671
17	205,02831
18	204,95195
19	204,92784
20	204,76711
21	204,64252
22	204,59029
23	204,42552
24	204,50188
25	204,5099
Promedio	204,674994
ErrorRelativo%	2,069380861

Tabla3. Resultados de las medidas de la celda de carga de 1 kg con calibración

Fuente: Elaboración propia

VI. RESULTADOS Y CONCLUSIONES

- El caudal que se encontró para la mini bomba de agua sumergible está dentro del rango especificado para este, lo cual sugiere que está funcionando correctamente y dentro de los parámetros esperados para su operación.
- La celda de carga de 1 kg con un factor de calibración de 248,8437002 presenta una buena precisión en las mediciones que realiza, ya que se determinó un error relativo porcentual de 2,069 %, este error no es muy grave, considerando a esta herramienta óptima para su uso en proyectos.

VII. BIBLIOGRAFÍA

- ALLDATASHEET. (2024). “Breve descripción de HX711”. En: <https://www.alldatasheet.es/view.jsp?Searchword=HX711>, (fecha de consulta 31/05/2024).
- AMAZON WEB SERVICES. (2023). “AWS IoT Analytics”. En: <https://aws.amazon.com/es/iot-analytics/>, (fecha de consulta 31/05/2024).
- AMAZON WEB SERVICES. (2023). “AWS IoT Core”. En: <https://aws.amazon.com/es/iot-core/?nc=sn&loc=0>, (fecha de consulta 10/05/2024).
- AMAZON WEB SERVICES. (2024). “AWS IoT Device Shadow service”. En: <https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html>, (fecha de consulta 10/05/2024).
- AMAZON WEB SERVICES. (2024). “Creating a Channel”. En: <https://docs.aws.amazon.com/iotanalytics/latest/userguide/create-channel.html>, (fecha de consulta 02/06/2024).
- AMAZON WEB SERVICES. (2024). “Creating a Data Store”. En: <https://docs.aws.amazon.com/iotanalytics/latest/userguide/create-data-store.html>, (fecha de consulta 02/06/2024).
- AMAZON WEB SERVICES. (2024). “Creating a Dataset”. En: <https://docs.aws.amazon.com/iotanalytics/latest/userguide/create-dataset.html>, (fecha de consulta 02/06/2024).

- AMAZON WEB SERVICES. (2024). “Creating a Pipeline”. En: <https://docs.aws.amazon.com/iotanalytics/latest/userguide/create-pipeline.html>, (fecha de consulta 02/06/2024).
- AMAZON WEB SERVICES. (2024). “¿Por qué Alexa?”. En: <https://developer.amazon.com/es-ES/alexa>, (fecha de consulta 31/05/2024).
- AMAZON WEB SERVICES. (2024). “Rules for AWS IoT”. En: <https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html>, (fecha de consulta 10/05/2024).
- HARRIS, Tom. (2024). “How Light Emitting Diodes (LEDs) Work”. En: <https://electronics.howstuffworks.com/led.htm>, (fecha de consulta 31/05/2024).
- IBM. (s.f.). “What is the Internet of Things (IoT)?”. En: <https://www.ibm.com/topics/internet-of-things>, (fecha de consulta 10/05/2024).
- NAYLAMP MECHATRONICS. (2023). “CELDA DE CARGA DE 1KG”. En: <https://naylampmechatronics.com/sensores/702-celda-de-carga-1kg.html>, (fecha de consulta 31/05/2024).
- NAYLAMP MECHATRONICS. (2023). “DRIVER PUENTE H L290N 2A DESCRIPCIÓN INFO”. En: <https://naylampmechatronics.com/drivers/11-driver-puente-h-l298n.html>, (fecha de consulta 10/05/2024).
- PORTAL ACADEMICO DEL CCH. (2021). “Tableta protoboard”. En: <https://portalacademico.cch.unam.mx/cibernetica1/implementacion-de-circuitos-logicos/tableta-protoboard>, (fecha de consulta 31/05/2024).
- ROBOCRAZE. (2022). “What is Ultrasonic Sensor: Working Principle Applications”. En: https://robocraze.com/blogs/post/what-is-ultrasonic-sensor?_pos=1&_psq=what+is+ultr&_ss=e&_v=1.0, (fecha de consulta 10/05/2024).
- TEJA, Ravi. (2024). “Getting Started with ESP32 | Introduction to ESP32”. En: <https://www.electronicshub.org/getting-started-with-esp32/>, (fecha de consulta 10/05/2024).
- UNIT ELECTRONICS. (2018). “Hoja de datos de Bomba de Agua sumergible Arduino”. En: <https://uelectronics.com/wp-content/uploads/2018/02/Hoja-de-datos-Bomba-de-Agua-sum.pdf>, (fecha de consulta 10/05/2024).
- W3SCHOOLS. (2024). “SQL Tutorial”. En: https://www.w3schools.com/sql/sql_

intro.asp, (fecha de consulta 02/06/2024).

VIII. ANEXOS

■ TIEMPO Y CAUDAL DE LLENADO DE UN VASO DE 130 ML CON MINI BOMBA DE AGUA

El tiempo promedio de llenado de un vaso de 130 ml con nuestra mini bomba de agua se calculo con la siguiente formula:

$$\overline{Tiempodelllenado} = \frac{\sum_{i=1}^n Tiempodelllenado}{n}$$

Donde n es la cantidad de tiempos registrados.

Con el tiempo promedio de llenado se procede a calcular el caudal en litros por hora con las siguiente formula:

$$Caudal = \frac{\frac{Volumen}{Tiempodelllenado} * 1}{\frac{1000}{1}} * 3600$$

$$Caudal = 80,83366918[L/H]$$

■ FACTOR DE CALIBRACIÓN ÓPTIMO

Antes de hallar el factor de calibración óptimo debemos presentar la formula del factor de calibración para cada observación:

$$Factordecalibracion = \frac{PesoDeSerialMonitor}{PesoReal}$$

Dando uso a la fórmula podemos encontrar el factor de calibración para cada una de las observaciones:

- $Factordecalibracion1 = \frac{52111}{209} = 249,3349282$
- $Factordecalibracion2 = \frac{52225}{209} = 249.8803828$
- $Factordecalibracion3 = \frac{52074}{209} = 249.1578947$
- $Factordecalibracion4 = \frac{51885}{209} = 248.2535885$
- $Factordecalibracion5 = \frac{52086}{209} = 249.215311$
- $Factordecalibracion6 = \frac{52019}{209} = 248.8947368$

- $Factordecalibracion7 = \frac{52158}{209} = 249.5598086$
- $Factordecalibracion8 = \frac{51445}{209} = 246.1483254$
- $Factordecalibracion9 = \frac{51981}{209} = 248.7129187$
- $Factordecalibracion10 = \frac{51924}{209} = 248.4401914$
- $Factordecalibracion11 = \frac{51729}{209} = 247.507177$
- $Factordecalibracion12 = \frac{52463}{209} = 251.0191388$

El factor de calibración óptimo de una celda de carga de 1 kg es igual al promedio de factores de calibración registrados, para obtener este resultado se uso la siguiente formula:

$$\overline{Factodecalibracion} = \frac{\sum_{i=1}^n FactorDeCalibracion}{n} = 248,8437002$$

Donde n es la cantidad de iteraciones en las que se obtuvo un factor de calibración.

■ **ERROR RELATIVO PORCENTUAL DE CELDA DE CARGA DE 1 KG CON CALIBRACIÓN**

Para calcular el error relativo porcentual de la celda de carga de 1 kg ya calibrada se uso la siguiente formula:

$$E \% = \left| \frac{PesoPromedio - PesoReal}{PesoReal} \right| \times 100$$

$$E \% = \left| \frac{204.674994 - 209}{209} \right| \times 100 = 2.06938$$